What is claimed is:

1. A method comprising:

allocating a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, a portion of the memory entry includes a unique identifier assigned to the instructions.

2. The method of claim 1, further comprising:

maintaining a count of threads included in the multithreaded engine that use the memory entry.

3. The method of claim 1, further comprising:

maintaining a bit to represent availability of the memory entry for thread use.

4. The method of claim 2 wherein maintaining the count includes incrementing the count to represent a thread initiating use of the memory entry.

5. The method of claim 2 wherein maintaining the count includes decrementing the count to represent a thread halting use of the memory entry.

6. The method of claim 3 wherein maintaining the bit includes setting the bit to represent availability of the memory entry for thread use.

7. The method of claim 3 wherein maintaining the bit includes clearing the bit to represent unavailability of the memory entry for thread use.

8. The method of claim 3, further comprising:

checking the bit to determine the availability of the memory entry for thread use.

9. The method of claim 1 wherein the unique identifier includes four bits.

10. The method of claim 1 wherein the memory entry identifies a location in a local memory included in the multithreaded engine of the packet processor.

11. A computer program product, tangibly embodied in an information carrier, the computer program product being operable to cause a machine to:

allocate a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, a portion of the memory entry

includes a unique identifier assigned to the instructions.

12. The computer program product of claim 11 being further operable to cause a machine to:

maintain a count of threads included in the multithreaded engine that use the memory entry.

13. The computer program product of claim 11 being further operable to cause a machine to:

maintain a bit to represent availability of the memory entry for thread use.

14. The computer program product of claim 12 wherein maintaining the count includes incrementing the count to represent a thread initiating use of the memory entry.

15. The computer program product of claim 12 wherein maintaining the count includes decrementing the count to represent a thread halting use of the memory entry.

16. The computer program product of claim 13 wherein maintaining the bit includes setting the bit to represent availability of the memory entry for thread use.

17. The computer program product of claim 13 wherein maintaining the bit includes clearing the bit to represent unavailability of the memory entry for thread use.

18. The computer program product of claim 13 being further operable to cause a machine to:

check the bit to determine the availability of the memory entry for thread use.

19. The computer program product of claim 11 wherein the unique identifier includes four bits.

20. The computer program product of claim 11 wherein the memory entry identifies a location in a local memory included in the multithreaded engine of the packet processor.

21. A memory manager comprises:

a process to allocate a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, a portion of the memory entry includes a unique identifier assigned to the instructions.

22. The memory manager of claim 21, further comprises:

a process to maintain a count of threads included in the multithreaded engine that use the memory entry.

23. The memory manager of claim 21, further comprises:

a process to maintain a bit to represent availability of the memory entry for thread use.

24. A system comprising:

a packet processor capable of,

allocating a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, a portion of the memory entry includes a unique identifier assigned to the instructions.

25. The system of claim 24 wherein the packet processor is further capable of:

maintaining a count of threads included in the multithreaded engine that use the memory entry.

26. The system of claim 24 wherein the network processor is further capable of:

maintaining a bit to represent availability of the memory entry for thread use.

27. A network forwarding device comprising:

an input port for receiving packets;

an output for delivering the received packets; and

a network processor capable of,

allocating a memory entry in a memory device to instructions executed on a multithreaded engine included in a packet processor, a portion of the memory entry includes a unique identifier assigned to the instructions.

28. The network forwarding device of claim 27, wherein the network processor is further capable of maintaining a count of threads included in the multithreaded engine that use the memory entry.

29. The network forwarding device of claim 28, wherein the network processor is further capable of maintaining a bit to represent availability of the memory entry for thread use.

30. A method comprising:

allocating a content-addressable-memory (CAM) entry to a microblock executed on a multithreaded microengine included in a network processor, a portion of the CAM entry includes a unique identifier assigned to the microblock.

31. The method of claim 30, further comprising:

maintaining a count of threads included in the multithreaded microengine that use the CAM entry.

32. The method of claim 30, further comprising:

maintaining a bit in a status register to represent availability of the CAM entry to identify a local memory location.